

FASE 2

# TECH CHALLENGE

SOFTWARE ARCHITECTURE

02 DE 04

## O QUE VEM POR AÍ

Tech Challenge é o projeto da fase que englobará os conhecimentos obtidos em todas as disciplinas da fase. Esta é uma atividade que, em princípio, deve ser desenvolvida em grupo. Importante atentar-se ao prazo de entrega, pois trata-se de uma atividade obrigatória, uma vez que vale 60% da nota de todas as disciplinas da fase.

## O PROBLEMA

Há uma lanchonete de bairro que está expandindo devido seu grande sucesso. Porém, com a expansão e sem um sistema de controle de pedidos, o atendimento aos clientes pode ser caótico e confuso. Por exemplo, imagine que um cliente faça um pedido complexo, como um hambúrguer personalizado com ingredientes específicos, acompanhado de batatas fritas e uma bebida. O atendente pode anotar o pedido em um papel e entregá-lo à cozinha, mas não há garantia de que o pedido será preparado corretamente.

Sem um sistema de controle de pedidos, pode haver confusão entre os atendentes e a cozinha, resultando em atrasos na preparação e entrega dos pedidos. Os pedidos podem ser perdidos, mal interpretados ou esquecidos, levando à insatisfação dos clientes e a perda de negócios.

Em resumo, um sistema de controle de pedidos é essencial para garantir que a lanchonete possa atender os clientes de maneira eficiente, gerenciando seus pedidos e estoques de forma adequada. Sem ele, expandir a lanchonete pode acabar não dando certo, resultando em clientes insatisfeitos e impactando os negócios de forma negativa.

Para solucionar o problema, a lanchonete irá investir em um sistema de autoatendimento de fast food, que é composto por uma série de dispositivos e interfaces que permitem aos clientes selecionar e fazer pedidos sem precisar interagir com um atendente, com as seguintes funcionalidades:



## Pedido

Os clientes são apresentados a uma interface de seleção na qual podem optar por se identificarem via CPF, se cadastrarem com nome, e-mail ou não se identificar, podendo montar o combo na seguinte sequência, sendo todas elas opcionais:

- / 1. Lanche
- / 2. Acompanhamento
- / 3. Bebida

Em cada etapa é exibido o nome, descrição e preço de cada produto.

## Pagamento

O sistema deverá possuir uma opção de pagamento integrada para MVP. A forma de pagamento oferecida será via QRCode do Mercado Pago.

## Acompanhamento

Uma vez que o pedido é confirmado e pago, ele é enviado para a cozinha para ser preparado. Simultaneamente deve aparecer em um monitor para o cliente acompanhar o progresso do seu pedido com as seguintes etapas:

- / Recebido
- / Em preparação
- / Pronto
- / Finalizado

## Entrega

Quando o pedido estiver pronto, o sistema deverá notificar o cliente que ele está pronto para retirada. Ao ser retirado, o pedido deve ser atualizado para o status finalizado.

Além das etapas do cliente, o estabelecimento precisa de um acesso administrativo:

## Gerenciar clientes

Com a identificação dos clientes o estabelecimento pode trabalhar em campanhas promocionais.

## Gerenciar produtos e categorias

Os produtos dispostos para escolha do cliente serão gerenciados pelo estabelecimento, definindo nome, categoria, preço, descrição e imagens. Para esse sistema teremos categorias fixas:

- / Lanche
- / Acompanhamento



**Acompanhamento de pedidos:** Deve ser possível acompanhar os pedidos em andamento e tempo de espera de cada pedido.

As informações dispostas no sistema de pedidos precisarão ser gerenciadas pelo estabelecimento através de um painel administrativo.

## Entregáveis

1



Atualizar a aplicação desenvolvida na FASE 1 refatorando o código para seguir os padrões clean code e clean architecture

a. Alterar/criar as APIs:

- / i. Checkout Pedido que deverá receber os produtos solicitados e retornar a identificação do pedido.
- / ii. Consultar status pagamento pedido, que informa se o pagamento foi aprovado ou não.
- / iii. Webhook para receber confirmação de pagamento aprovado ou recusado.
- / iv. A lista de pedidos deverá retorná-los com suas descrições, ordenados com a seguinte regra:
  - / 1. Pronto > Em Preparação > Recebido;
  - / 2. Pedidos mais antigos primeiro e mais novos depois;
  - / 3. Pedidos com status Finalizado não devem aparecer na lista.
- / v. Atualizar o status do pedido.
- / vi. Como desafio extra, opcionalmente, você pode implementar a integração com Mercado Pago para gerar o QRCode para pagamento e integrar com o WebHook para capturar os pagamentos. Caso contrário, será necessário realizar o mock da parte de pagamentos. Como referência, acesse: **site do mercado pago**.

2



Criar uma arquitetura em Kubernetes que atenda os seguintes requisitos

a. Os requisitos funcionais descritos nos itens anteriores (item problema).



c. Os arquivos manifestos (yaml) precisam estar no Github junto com a nova versão do código.

3



Entrega da seguinte documentação no ReadMe

a. Desenho da arquitetura pensado por você, pessoa arquiteta de software, contemplando:

- / i. Os requisitos do negócio (problema).
- / ii. Os requisitos de infraestrutura:
  - / 1. Você pode utilizar o MiniKube, Docker Kubernetes, AKS, EKS, GKE ou qualquer nuvem que você desenha.

b. Collection com todas as APIs desenvolvidas com exemplo de requisição (que não seja vazia):

- / i. Link do Swagger no projeto ou link para download da collection do Postman (JSON).

c. Guia completo com todas as instruções para execução do projeto e a ordem de execução das APIs, caso seja necessário.

d. Link para vídeo demonstrando a arquitetura desenvolvida na nuvem ou localmente

- / i. O vídeo deve ser postado no Youtube ou Vimeo.
- / ii. Não esqueça de deixá-lo público ou não listado.

No arquivo entregue na plataforma, é necessário somente colocar a URL do Github com as informações.

02 DE 04

VÍDEO 01 DE 01

PLAYLIST

01 DE 01

AUTOPLAY

01

Boas-vindas F2  
Arquitetura de...

Em

02:58

progresso



TRANSCRIÇÃO



## CONTE-NOS SOBRE A SUA EXPERIÊNCIA

O que você achou do conteúdo deste capítulo?

